

Relocation Framework to Improve the Usage Efficiency of Bicycle-sharing Systems

Chun-Hee Lee, Jeong-Woo Lee, Yung-Joon Jung and Il-Yeon Cho

Artificial Intelligence Research Laboratory, ETRI

218 Gajeong-ro, Yuseong-gu, Daejeon, 34129, SOUTH KOREA

{ch.lee, jeongwoo, jjing, iycho}@etri.re.kr

Abstract— Recently, a bicycle-sharing system has been greatly spotlighted as a second transportation vehicle. However, since shared bicycles are used for tons of users, the bicycles should be relocated accordingly. To handle the bicycle relocation problem, relocation managers move bicycles from one station to another station by using their own experiences. Relocating bicycles by the managers' experiences might be ineffective and inconsistent.

In this paper, we propose an effective and systematic relocation framework which consists of the demand forecasting step and the relocation step. In the demand forecasting step, we try to precisely forecast the bicycle demand by utilizing multiple machine learning techniques. In the relocation step, we provide an algorithm to optimize the relocation process. We are expecting that the relocation process will be greatly improved if our relation framework is used in the real bicycle-sharing system.

Keywords— Bicycle-sharing System, Demand Forecasting, Bicycle Relocation, Regression, LSTM

I. INTRODUCTION

Recently, a bicycle-sharing system as a second transformation vehicle has been spotlighted because of several reasons such as health and environmental issues. Many local governments are trying to introduce a bicycle-sharing system and promote the use of the system.

However, because shared bicycles can be used by many users simultaneously, there is a case that users cannot rent a bicycle when they want to ride. If rentals and returns by users occur in a balanced way, we do not need to relocate bicycles. The problem is that such cases never occur in the real world. Generally, bicycle demands by users are biased. For example, during rush hours in the morning, bicycles are usually returned in an industrial area while they are returned in a residential area during rush hours in the evening. Therefore, to avoid users' inconvenience, relocating bicycles should be done properly.

To solve this relocation problem, relocation managers move bicycles everyday by using their own experiences. Because it is not a systematic way, it might be ineffective and inconsistent. Also, it is difficult to transfer the relocation job to a new relocation manager if an experienced relocation manager quits the job. In this paper, we propose an effective and systematic relocation framework to improve the efficiency

of relocating bicycles. Our proposed framework has two major steps as follows:

- **Demand Forecasting Step:** In this step, we forecast the bicycle demand for each station. It is important to forecast the bicycle demand. If we forecast the demand precisely, we can figure out how we relocate bicycles. For more accurate forecasting, we apply different models to each station and time zone by considering multiple machine learning models. To find the best model for each station and time zone, we propose a model selection algorithm.
- **Relocation Step:** Based on the results from the demand forecasting step, we propose an algorithm to optimize the relocation process to get the best benefit. Since the results of demand forecasting might have errors, we have to consider the errors when the relocation process is optimized. To overcome the errors of demand forecasting, we introduce MIN_BIKE which is the minimum number of bicycles for a station to keep.

Some research works have been proposed to handle the bicycle relocation problem [1, 2, 3]. The papers in [1, 2] used a similar framework to ours, which consists of the forecasting step and the relocation step. Lee and Son [1] dealt with the relocation problem by focusing on the case of the Seoul bicycle-sharing system. However, to forecast the number of bicycles, they used simple equations while we use multiple machine learning models to improve the accuracy of demand forecasting. Also, in [2], to forecast the bicycle demand, the models for the pick-up demand and the drop-off demand were built. The pick-up demand was calculated using a Meteorology Similarity Weighed K-Nearest-Neighbor (MSWK) regressor and the drop-off demand was calculated using an inter station bike transition (ISBT) predictor. However, in this paper, we consider multiple machine learning models and one of them is LSTM (Long Short-Term Memory) [4], which is a deep learning model. It is well known that LSTM is very good for learning complex time series data. The author in [3] used LSTM for forecasting the number of bicycles and mentioned the relocation problem but did not propose a detailed relocation strategy.

The rest of the paper is organized as follows. In Sections 2 and 3, we describe demand forecasting methods and the

relocation framework in detail. In section 4, data preparation, implementation and future work will be discussed. We finally conclude the paper in Section 5.

II. DEMAND FORECASTING IN A BICYCLE-SHARING SYSTEM

Bicycles in a bicycle-sharing system can be rent and returned only on the designated bicycle stations in many cases. If the rental and return pattern of each station is very similar, we can forecast the pattern by using one forecasting model. However, each station is deployed in a completely different circumstance and the characteristics of main users in each station might be different. Therefore, the rental and return pattern of each station shows heterogeneity. For example, see two bicycle stations of Sejong City in Korea as shown in Figure 1. Figure 1 shows the rental patterns of Sejong government building and Hosu park between January 2020 and April 2020. The x-axis represents time (from 0H to 23H) and the y-axis represents the number of rentals. The peak time is between 2 and 4 PM in Hosu park since most of rentals are used for recreation purposes while the peak time is the rush hours and the lunch time near the government building.

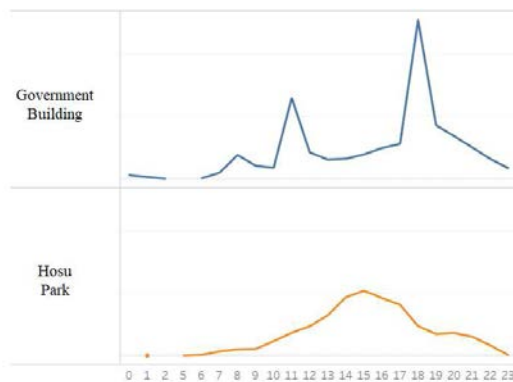


Figure 1. Rental Patterns in Two Different Stations

To learn rental and return patterns effectively, we build a demand forecasting model according to each station. Also, the utilization of multiple machine learning models might be more effective rather than that of a single machine model. Therefore, we can consider 3 types of machine learning models for exploiting different characteristics of patterns. The advantages and the disadvantages of models are summarized in Table 1.

TABLE 1. ADVANTAGES AND DISADVANTAGES OF MODELS

Model	Advantages	Disadvantages
Statistics Model	- The amount of computation is low - The explainability is high	- It is not easy to fit complex patterns
Regression Model	- It is simple but well used for many areas - The explainability is medium	- It is difficult to exploit time series patterns
LSTM Model	- It is good to exploit complex time series	- The amount of computation is high

	patterns	- It needs a lot of space to save a model
--	----------	---

The bicycle demand by users means the number of bicycles the users will rent in the future. However, some users might return bicycles. In the point of view of relocation, we should consider the effect of returned bicycles by users. Because of that reason, we try to forecast “the number of rent bicycles” – “the number of returned bicycles.” instead of “the number of rent bicycles.”

First of all, we consider a simple statistics model. Some of bicycle stations might show regularity according to the time and the day of the week since some users ride shared bicycles on a regular basis. In that case, we do not need to maintain heavy machine learning models. It is good to build a statistics model for that station. The statistics model can be built in a fast and simple way. Our statistics model can be constructed as follows:

- Cluster data according to the day of the week and the hour
- Compute the frequent item for each cluster
- Make a table of statistics with keys (day of the week, hour) and the value (bicycle demand)

As summarized in Table 1, the statistics model is fast to build but not easy to represent complex patterns. Because of that, we consider a regression model as another model such as XGBoost and MLP Regressor. The regression model can fit more complex patterns than the statistics model. To fit the regression model better, we can add external data. For example, the weather significantly impacts on the rental/return patterns of shared bicycles. Thus, we add the weather data to fit the regression model better. Also, current bicycle demands can be affected by the previous status. Therefore, we add the historical data to the model. Our regression model has the following independent variables.

- Variables Related to Time: day of the week, hour, holiday
- Variables Related to Weather: temperature, wind direction, wind speed, rain, pressure, humidity
- Historical Data: the number of rentals and returns in (t-1) step, the number of rentals and returns in (t-2) step, where t is the current step

We use the MLP (Multi-Layer Perceptron) regression model among various regression models. However, it can be easily replaced with other regression models [5].

Finally, we use LSTM (Long Short-Term Memory) [4]. LSTM is based on RNN (Recurrent Neural Network). It is well known that LSTM fits very well for complex time-series data. To fit the LSTM model, we use additional data like the regression model as follows:

- Variables Related to Time: day of the week, hour, holiday
- Variables Related to Weather: temperature, wind direction, wind speed, rain, pressure, humidity

We can skip the historical data to fit the LSTM model because the LSTM model can train historical data by sliding windows. Based on three models, we propose an algorithm to choose the best model for each station as follows:

Algorithm 1: Selecting the best model for each station

1: For all the bicycle stations, train statistics models
 2: Assign the statistics model to the station whose accuracy is more than the threshold
 3: For the remaining stations, train regression models
 4: Assign the regression model to the station whose accuracy is more than the threshold
 5: For the remaining stations, train LSTM models
 6: Assign the LSTM model to the station whose accuracy is more than the threshold
 7: For the remaining stations, compare the accuracies of statistics model, regression model, LSTM model and assign the model with the highest accuracy

The reason why we choose the model like Algorithm 1 is trade-off between time and accuracy. Even if LSTM can represent more complex patterns but it is not easy to train and keep the model. However, the statistics model is very simple and fast to train the model. Therefore, for the first time, we apply a simple and fast model to all the bicycle stations. After checking a simple and fast model, we adopt a more complex model.

The rental and return patterns show heterogeneity according to stations as well as time zones. Therefore, we extend Algorithm 1 to consider both stations and time zones (0H ~ 23H). The process to consider both is similar to Algorithm 1.

III. RELOCATION FRAMEWORK IN A BICYCLE-SHARING SYSTEM

In this section, we will propose an effective and systematic relocation framework based on the demand forecasting models in the previous section. We develop a relocation framework with the following assumptions:

- It is assumed that the cost of driving a truck is 0
- The number of bicycles to be loaded in a truck is not limited

Also, we consider two measures to estimate the performance of the relocation framework.

- The number of shortage stations: The shortage station means the station whose parked bicycles are 0. We have to fill bicycles in that station.
- The number of relocated bicycles: It is the number of bicycles to be relocated by the relocation manager. Because we assume that the driving cost is 0, if we move N bicycles from one station to another station, the number of relocated bicycles is N.

In this paper, we invent an algorithm to relocate bicycles to minimize the number of shortage stations given the maximum

number of relocated bicycles τ . To perform the relocation algorithm, we first forecast the bicycle demand for each station (Line 3). Note that the bicycle demand means “the number of rent bicycles” – “the number of returned bicycles”. Next, we evaluate the number of bicycles to be collected for relocation in each station (Lines 4-11). Since each station might have the demand in the future, we collect the bicycles from only stations having spare bicycles. To evaluate the spare bicycles, we compute the number of parked bicycles in the next time ($next_bike_num(i)$) (Lines 4-5). In the example of Figure 2, $next_bike_num()$ is $4 - (-2) = 6$ (Station 1), $1 - 2 = -1$ (Station 2), and $0 - 1 = -1$ (Station 3). Based on $next_bike_num()$, the number of spare bicycles is evaluated

$$next_bike_num(i) - MIN_BIKE$$

If $next_bike_num(i) - MIN_BIKE < 0$, station i has no spare bicycles. MIN_BIKE is the number of allowable bicycles to avoid the forecasting error. We can usually set MIN_BIKE to 1. Because the number of relocations is limited, we sort stations according to $next_bike_num(i)$ in a descending order. Then we stop collecting bicycles if the number of collected bicycles is over the given threshold τ . Also, if there can be a case that the number of remaining relocations is less than the spare bicycles, the number of bicycles to be collected ($collection_bike_num$) is computed as follows:

$min(next_bike_num(i) - MIN_BIKE, rem)$, where rem is the number of remaining relocations.

In Figure 2, MIN_BIKE is set to 1. Let's evaluate spare bicycles for each station.

- Station 1: $6 - 1 = 5$
- Station 2: $-1 - 1 = -2$
- Station 3: $-1 - 1 = -2$

Only Station 1 has 5 bicycles as the number of spare bicycles. Therefore, we collect 5 bicycles in Station 1.

Next, we distribute spare bicycles to stations that need more bicycles. The number of bicycles to be distributed is evaluated as follows:

$$MIN_BIKE - next_bike_num(i)$$

We distribute collected bicycles ($num_collect$ in Line 12) to only stations such that $MIN_BIKE - next_bike_num(i) > 0$. The distribution process (Lines 13-16) is similar to the collection process (Lines 6-11). In the example of Figure 2, 2 bicycles and 2 bicycles are distributed to Station 2 and Station 3, respectively.

Even though the number of bicycles to be collected or distributed for each station was determined, we did not determine how we move bicycles from some station to another station yet. In order to do that, we make the OD table, which records how some bicycles are relocated. To make the OD table, we choose station S whose $collection_bike_num()$ is more than 0 (Line 17). Next, we have to distribute $collection_bike_num(S)$ bicycles to other bicycle stations.

Because it is effective to drop by the near station first, we first check the nearest station T (Line 19). If $\text{distribution_bike_num}(T) > 0$, we distribute bicycles to station T accordingly (Lines 21-24). And we save the relocation record $\langle S, T, N \rangle$ into OD table. $\langle S, T, N \rangle$ means we should relocate N bicycles from station S to station T .

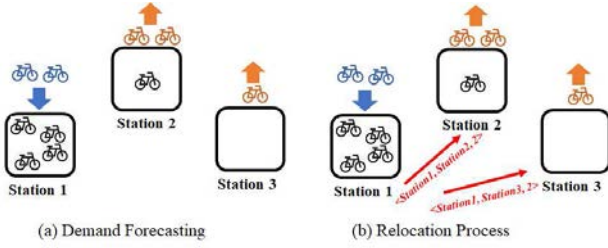


Figure 2. Relocation Example

Algorithm 2: Relocation algorithm

```

1:  $rem = \tau$ 
2:  $num\_collect = 0$ 

# demand forecasting using the models in Section 3
3: forecast bicycle demand for each station  $i$  ( $pred(i)$ )

4: for each station  $i$ 
5:    $next\_bike\_num(i) = curr\_bike\_num(i) - pred(i)$ 

# evaluation of the number of bicycles to be collected
6: for each station  $i$  # descending order in  $next\_bike\_num$ 
7:   if  $rem > 0$ 
8:     if  $next\_bike\_num(i) > MIN\_BIKE$ 
9:        $collection\_bike\_num(i) =$ 
          $\min(next\_bike\_num(i) - MIN\_BIKE, rem)$ 
10:       $num\_collect = num\_collect + collection\_bike\_num(i)$ 
11:       $rem = rem - collection\_bike\_num(i)$ 

# evaluation of the number of bicycles to be distributed
12:  $rem = num\_collect$ 
13: for each station  $i$  # ascending order in  $next\_bike\_num$ 
14:   if  $rem > 0$  and  $next\_bike\_num(i) < MIN\_BIKE$ 
15:      $distribution\_bike\_num(i) =$ 
        $\min(MIN\_BIKE - next\_bike\_num(i), rem)$ 
16:      $rem = rem - distribution\_bike\_num(i)$ 

# relocation process
17: for each station  $S$  where  $collection\_bike\_num(S) > 0$ 
18:    $coll = collection\_bike\_num(S)$ 
19:   for each station  $T$  in neighbor stations of  $S$ 
     (sorting by the distance in ascending order)
20:     if  $coll == 0$  then break
21:     if  $distribution\_bike\_num(T) > 0$ 
22:        $N = \min(distribution\_bike\_num(T), coll)$ 
23:        $distribution\_bike\_num(T) = distribution\_bike\_num(T) - N$ 
24:        $coll = coll - N$ 

```

25: save (S, T, N) to OD Table

IV. DATA PREPARATION, IMPLEMENTATION AND FUTURE WORK

To validate the effectiveness of our relocation framework, we are planning to do the virtual experiment by simulating the bicycle-sharing system in Sejong-City in Korea [6] and analyze the results in detail. We will assume that relocation managers can relocate bicycles every hour and the relocation framework can rebuild machine learning models every day. To do the virtual experiment, we collected the rental and return records in the Sejong bicycle-sharing system by collaborating with Sejong public officials. Since the raw data has duplicate data and inconsistent data, we cleaned the raw data. And, we downloaded the weather data in Sejong-City from Korea Meteorological Administration [7]. We are implementing our framework using Python. Especially we are using Pandas, Scikit-Learn and Keras libraries. Pandas is being used for cleaning data and Scikit-Learn and Keras are being used for learning regression model and LSTM, respectively.

In the future, we will deploy our relocation framework to the bicycle-sharing system in Sejong. Also, we will conduct extensive experiments according to models and parameters.

V. CONCLUSION

In this paper, we handled the bicycle relocation problem. To solve the problem, we proposed an effective and systematic relocation framework which consists of the demand forecasting step and the relocation step. In the demand forecasting step, we invented a method to improve the accuracy of demand forecasting models by considering multiple machine learning models. Also, we proposed an algorithm to optimize the relocation process.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2018-0-00225, Development of City Interior Digital Twin Technology to establish Scientific Policy)

REFERENCES

- [1] E. Lee and B. Son, "Optimal Rebalancing Strategy for Public Bike-sharing System in Seoul," *Journal of Korean Society of Transportation*, pp. 27-38, February 2019.
- [2] J. Liu, L. Sun, W. Chen, and H. Xiong, "Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization," *SIGKDD Conference on Knowledge Discovery and Data Mining*, August 2016.
- [3] Y. Soh, "Dynamic Inventory Level Management of Bike-Sharing System with LSTM," *master's thesis. Seoul National University*, 2018.
- [4] S. Hochreiter and J. Schmidhuber, "LONG SHORT-TERM MEMORY," *Neural Computation* vol. 9, no. 8, pp. 1735-1780, 1997.
- [5] C. M. Bishop, "Pattern Recognition and Machine Learning," *Springer*
- [6] <https://www.sejongbike.kr/>
- [7] <https://data.kma.go.kr/>



Chun-Hee Lee is a senior researcher at ETRI (Electronics Telecommunications Research Institute). He received the PhD degree in computer science from KAIST (Korea Advanced Institute of Science and Technology), in 2010. His research interests include large-scale data processing and simulation.



Jeong-Woo Lee received the B.S. degree in information and telecommunication engineering from Jeonbuk National University, Jeonju, Korea, in 1996, and the M.S. degree in information and communications engineering from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 1998. He received the Ph.D. degree in the Information and Communications Department from GIST in 2003. He is currently working in Electronics and

Telecommunications Research Institute (ETRI). His research interests include digital video coding algorithms, gpu-based coding algorithms, bigdata analysis and ML-based estimation algorithms.



Yung-Joon Jung received the B.B. degree in Physics from, Hankuk University of Foreign Studies, Korea in 1997, received the MS degree in Computer Science from same University in 1999 and received Ph. D degree in Computer Science from Chungnam National University in 2016. Since 2001, he has been with Electronics and Telecommunications Research Institute (ETRI), Korea, as a Principal Researcher. His research

interests are embedded operating system, real-time distributed computing, power management system, digital twin data analytics and software simulation.



Il-Yeon Cho received his BS and MS in industrial engineering from Sungkyunkwan University in 1991 and 1993, respectively. He received his PhD in computer engineering from Chungnam National University in 2007. He is currently a principal researcher at ETRI, Daejeon, Rep. of Korea. His research interests include system software, embedded systems.